International Journal of Pure and Applied Mathematics
Volume 120 No. 6 2018, 211-222
ISSN: 1314-3395 (on-line version)
url: http://www.acadpubl.eu/hub/
Special Issue

http://www.acadpubl.eu/hub/

DESIGNING OF LOW POWER HIGH SPEED FAST FOURIER TRANSFORM (FFT) PROCESSOR

Mr.U.Rajender[1],
Assistant professor
Dept of ECE
Vaageswari college of engineering
Karimnagar
urajender10@gmail.com

October 10, 2018

Abstract

Fast Fourier transform (FFT) is an efficient algorithm to calculate Discrete Fourier Transform (DFT) and its inverse. A wide variety of applications like Digital Signal processing and image processing rely heavily on it. The FFT computation is done by the FFT processors and its design is a key factor for the application. The proposed design implements a radix-4 FFT processor, which incorporates a low power commutator and a butterfly structure without a multiplier. The parallel pipe lined architecture of the processor also has higher throughput with lowered power consumption.

***Key Words*:** dragonfly structure; shift addition; commutator

1 INTRODUCTION

The conventional method of Fast Fourier Transform FFT calculation involves N^2 complex multiplications and N(N-1) complex addi-

tions. The radix-2 Cooley-Tukey algorithm performs the same computation involving (N/2)log_2^N complex multiplications and (N)log_2^N complex additions. So, a 16point FFT requires 256 multiplications and 240 additions. It is reduced to 64 multiplications and 192 additions when the proposed radix-4 approach is used. The equations for radix-4 FFT are:

$$X(4k) = \sum_{n=0}^{\frac{N}{4}-1} \{x(n) + x\left(n+\frac{N}{4}\right) + x\left(n+\frac{N}{2}\right) + x\left(n+\frac{3N}{4}\right)\} W_N^0 \; W_{N/4}^{Kn}$$

$$X(4K+1) = \sum_{n=0}^{\frac{N}{4}-1} \{x(n) - j\,x\left(n+\frac{N}{4}\right) - x\left(n+\frac{N}{2}\right) + j\,x\left(n+\frac{3N}{4}\right)\} W_N^n \; W_{N/4}^{Kn}$$

$$X(4K+2) = \sum_{n=0}^{\frac{N}{4}-1} \{x(n) - x\left(n+\frac{N}{4}\right) + x\left(n+\frac{N}{2}\right) - x\left(n+\frac{3N}{4}\right)\} W_N^{2n} \; W_{N/4}^{Kn}$$

$$X(4K+3) = \sum_{n=0}^{\frac{N}{4}-1} \{x(n) + j\,x\left(n+\frac{N}{4}\right) - x\left(n+\frac{N}{2}\right) - j\,x\left(n+\frac{3N}{4}\right)\} W_N^{3n} \; W_{N/4}^{Kn}$$

A. 16 point 2-parallel FFT design:

The input data stream was split into even and odd data streams which were then sent to two commutators. The output from the commutator was then fed in to the butterfly unit. The butterfly unit computes the factors which are then multiplied with co-efficient in the multiplier unit. A shuffle was needed to reorder the outputs from the butterfly stage1, so that the values were stored in the intermediate registers and then fed to butterfly stage2. The output was stored in the same commutator.

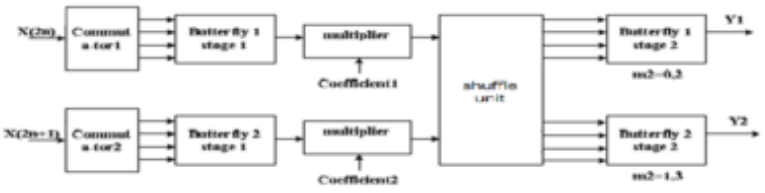

Figure 1 2-parallel pipeline architecture

B. Power saving commutator

In real time applications, the input is in the form of a sequential stream so a commutator is required to reorder the input. The commutator was found to be more power efficient as its architecture uses Dual port Random Access Memory RAM (DR) that involves no switching activity. The previous architecture used Shift registers (SR) the shifting increased switching activity thereby increasing the power consumption ultimately

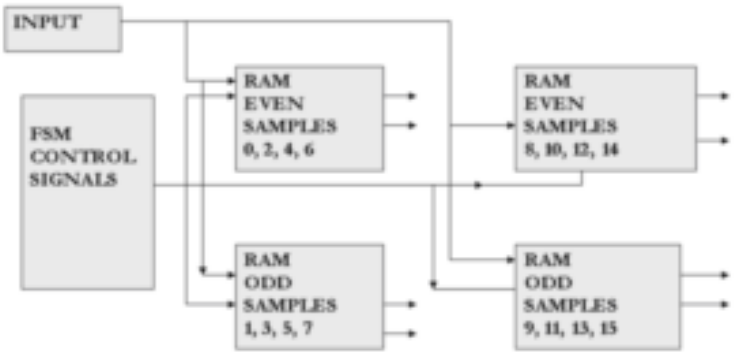

Figure 2 Commutator Block diagram

C. Low power butterfly design

Low power butterfly architecture based on 2s complement employed reduces the number of adders and sub tractors. The input samples groups e.g. 0,4,8,12 were each required 4 times in the first stage computation, these computations were done separately in the previous schemes requiring 8 clock cycles now the proposed scheme computes the 4 equations simultaneously, so it required only 2 clock cycles. The speed therefore is drastically increased.

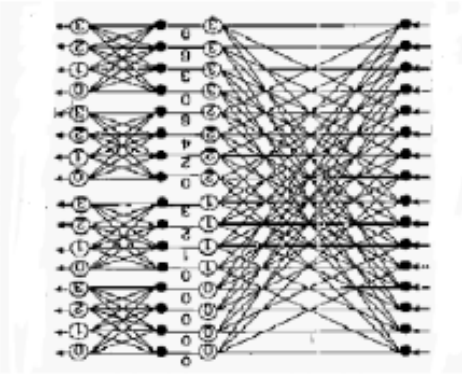

Figure 3 Signal flow graph radix-4 DIT-FFT

Table 1 Butterfly Dragonfly Equations

K Value	Butterfly equation for radix-4 DIT
0	(ar+cr)+(br+dr)+J[(ai+ci)+(bi+di)]
1	(ar-cr)+(bi-di)+j[(ai-ci)-(br-dr)]
2	(ar+cr)-(br+dr)+j[(ai+ci)-(bi+di)]
3	(ar-cr)-(bi-di)+j[(ai-ci)+(br-dr)]

Table 2 Twiddle coeffiecnts for Butterfly Equations

Wn	Decimal value	Original quantized real	Originalquantized imaginary
W0	1	7fff	0000
W1	1	7fff	0000
W2	0.707-0.707j	5a82	a57d
W3	0.383-0.924J	30fb	89be
W4	0-j	0000	8000
W6	-0.707-0.707j	a57d	a57d
W9	-0.9238-(-0.38268)j	89be	30fb

D. Simple Multiplier less design for a Multiplier

The input to be multiplied was D-multiplexed and given to two shift-add modules. The output from each Shift-add module can be added or subtracted and also multiplexed to get the output. The

input can also be swapped to obtain complex conjugates whenever necessary.

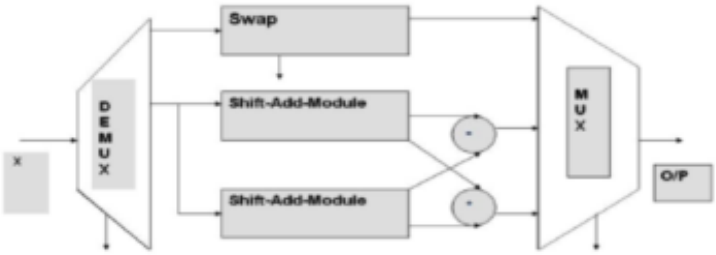

Figure 4 Block Diagram of Multiplier

E. Shift Add module of a Multiplier

The block diagram of the shift adds modules shown in Figure 5 performs multiplication by shift and add method. The computation is only done here

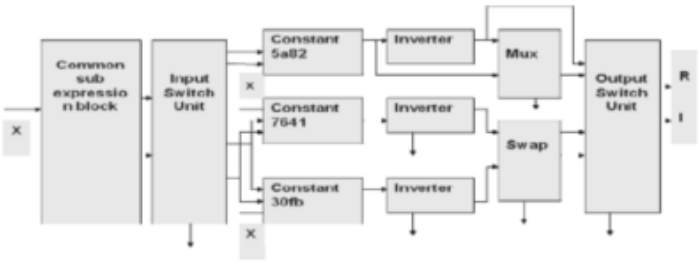

Figure 5 Block Diagram of a Shift Add Unit

The multiplication of a value x with 5a82 was realized by first left shifting it by 2 bits and then adding the shifted value to x. This is equivalent to multiplying with 5. Similarly 65x was obtained by first left shifting it by 6 bits an then adding the shifted value to x. These functions were done by the common sub expression block. The value 5a82x was obtained by adding values obtained by left shifting 5x 12 times, left shifting 5x 9 times and the value obtained by left shifting 65x once.

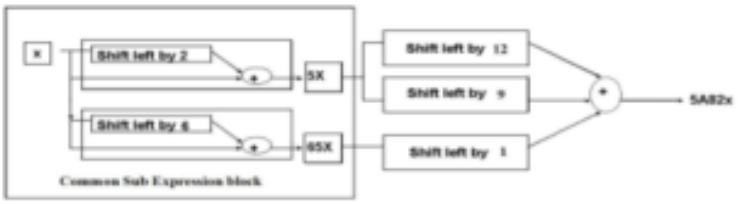

Figure 6 Multiplication with 5a82 Realised with Shift Addition

Similarly, the multiplications with other twiddle factors 7641 and 30fb were implemented as shown in figure 7 and 8.

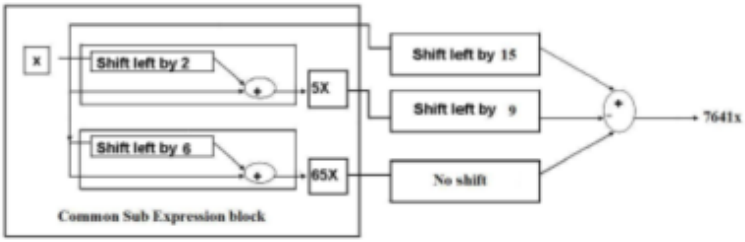

Figure 7 Multiplication with 7641 Realised with Shift Addition

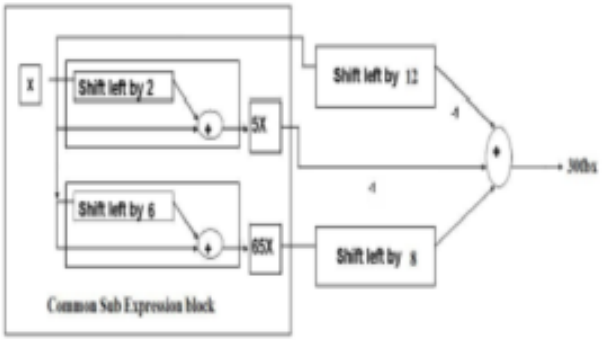

Figure 8 Multiplication with 30fb Realised with Shift Addition

2 SIMULATION RESULTS

The FFT core was simulated in Modelsim tool in Verilog HDL and synthesized by Xilinx ISE 11.7 tool for fixed point(fp) with 64(32 real, 32 imaginary) bits and same results were verified with Matlab

A. Commutator

FFT of 16 samples were computed, the commutator used 4 DR each with 4 registers. The function of the commutator is to collect, store and reorder the samples of input data for further processing in the butterfly stages. This was done by using a Finite State Machine (FSM) that generated the control signals to select the even and odd RAMs. The chip select lines selected the even and odd RAM. The address lines selected the memory location for storing the samples. Here the samples when received were stored in the order they were required for computation in the butterfly stage. This reduced a number of registers which were needed for re-ordering in the previous architectures, thereby reducing the size and computational complexity.

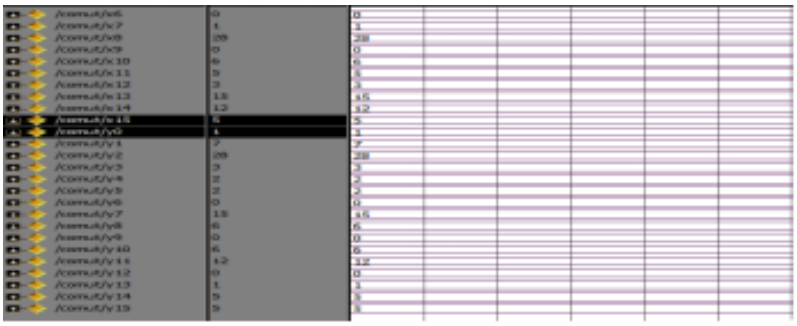

Figure 9 Simulation result of Commutator

B. Butterfly Stage

The butterfly expressions shown (figure3) are implemented here. Adders and inverters were used to get output (a + c) (a - c), (b + d) , -(b + d), (b d) , - (b d) . The real and imaginary equations were computed simultaneously and in parallel with just inputs a, b, c, d. The output from stage1 was then multiplied with

the twiddle co-efficient, shuffled using a shuffling unit and then sent to stage2 that calculated the equations in a similar fashion.

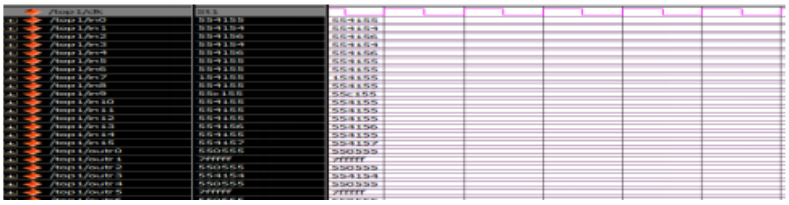

Figure 10 Simulation result of Butterfly Stage1

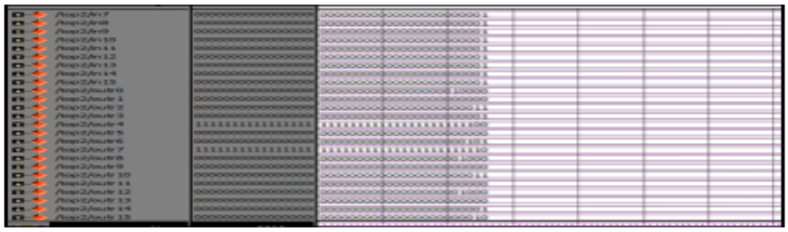

Figure 11 Simulation result of Butterfly Stage2

C. Shift Add Module

Block diagram shown in figure5 was implemented. The other twiddle factors were obtained from the 3 main factors 5a82, 7641, 30fb. So factors a57d, 89be, cf04 were all obtained by inverting 5a82, 7641, 30fb. The factors were then swapped as and when needed. The input and output were controlled using switches.

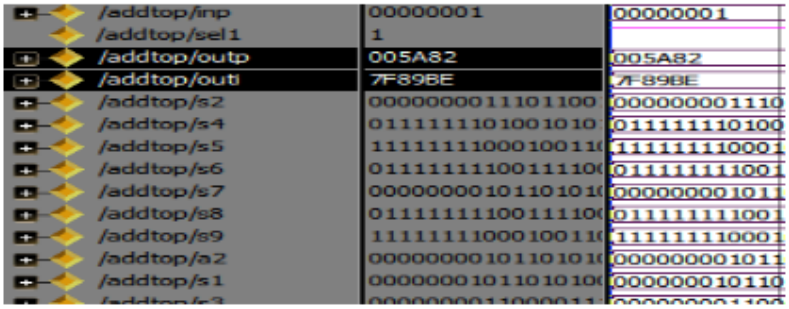

Figure 12 Simulation result of Shift Add Module

SYNHESIS RESULTS

The proposed design of 16-point FFT was synthesized in altera quartus tool by this tool we can calculate power , speed and area. RTL Compiler targeting the TSMC 0.18 CMOS technology library. The RTL block diagram for complete FFT module is

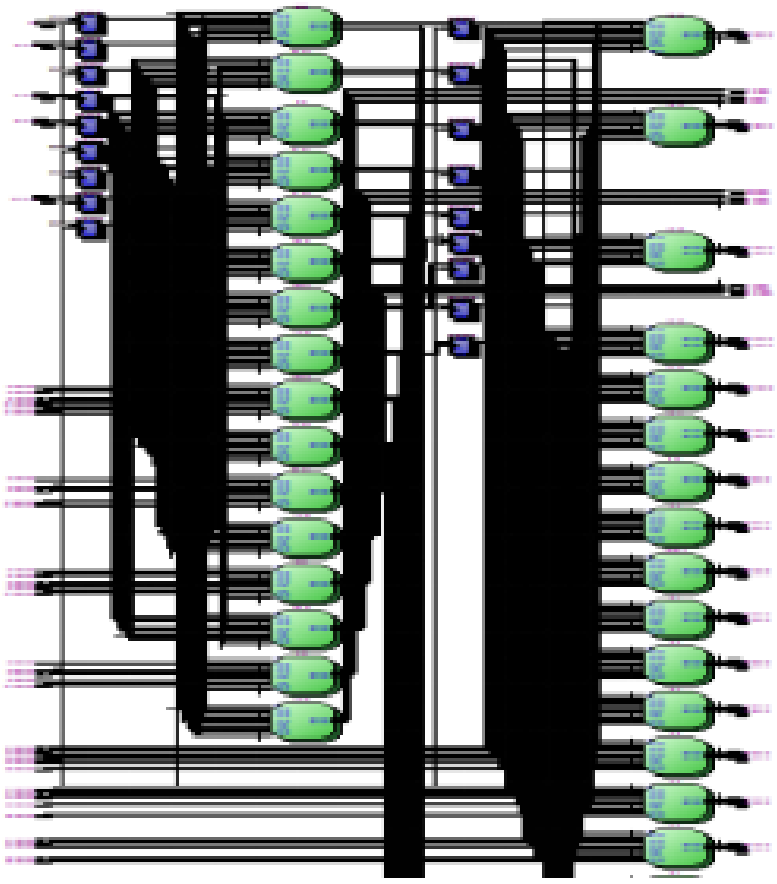

Figure 13 RTL Schematic for 16-point FFT

a. Power consumption

The fixed point FFT module consumed a total power of 146.74mw

Table 3 Power consumption in radix-4 fixed point

Type of power dissipation	Value
Dynamic thermal power dissipation	0.00mw
Static thermal power dissipation	80.05mw
I/O thermal power dissipation	66.69mw
Total Power dissipation	146.74mw

Table 4 Power Consumption of Different module

Module Name	Value of power dissipation
Commutator	33.3mw
Shift Add unit	56.78mw
Multipier	78.mw
Butterfly Unit	84.3mw

From table 4 Commutator consuming less power when compare to the all modules. The power consumption of fixed point FFT can be decrese compare to the other architectures.

A. Area

The shift add unit consumed a lesser power and an area of 8761m which is less when compared to the conventional multiplier unit and the total logical elements required is 1254 . So, compared to the architectures this architecture consists less area.

B. Speed

The FFT core computed with a speed of 364.17 MHz which is faster than 250MHz.

3 CONCLUSION

The parallel pipelined architecture for 16 point radix-4 DIT FFT in fixed point representation was proposed and implemented. The power reduction method was implemented at every stage to obtain maximum power savings. The features were

- High-performance 16-point Complex FFT
- Two's Complement Arithmetic
- Flexible I/O And Memory Configurations
- Naturally Ordered Input And Output Data
- Parallel Pipelined Processor
- Maximum Speed Up To 364.17 MHz.
- Low Power Processor

References

[1] Cooley, J.W. and J.W. Tukey, 1965. An algorithm for the machine computation of the complex Fourier series. Math. Computation, 19: 297-301.

[2] Han, W.T. Arslan, A.T. Erdogan and M. Hasan, 2005. Low power commutator for pipelined FFT processors. IEEE, pp: 5274-5277.

[3] Han, W.T. Arslan, A.T. Erdogan and M. Hasan,2005. The development of high performance FFT IP cores through hybrid low power algorithmicmethodology.IEEE,pp:549-552.

[4] Rabiner, L.R. and B. Gold, 1975. Theory and Application of Digital Signal Processing Prentice-Hall.

[5] John, G.P. and D.G. Manolakis, 1988. Introduction to Digital Signal Processing. Mac Millian.

[6] Hasan, M. and T. Arslan, 2003, 2003. A triple port RAM based low power commutator architecture for a pipelined FFT processor. Circuits and Systems, ISCAS03. Proc. Intl. Symp., 5: V-353 - V-356.

[7] Han, W.T. Arslan, A.T. Erdogan and M. Hasan, 2004. A novel low power pipelined FFT based on sub expression sharing for wireless LAN applications. IEEE Workshop on Signal Processing Systems, pp: 83-8